VIDEO SOURCE CODING WITH SIDE INFORMATION

Technical Field

The invention is concerned with video coding and, more particularly, with coding techniques utilizing side information.

Background of the Invention

Incorporated herein by reference is the patent application of Rohit Puri and Kannan Ramchandran, "Encoding and Decoding of Digital Data Using Cues Derivable at a Decoder", U.S. Application No. 10/651,854, filed on August 29, 2003, wherein related techniques are described.

Video compression algorithms predicated on source coding with side information (also known as distributed source coding) are becoming increasingly popular, offering attractive features such as:

(1) Flexible distribution of total codec complexity between the encoder and the decoder in contrast to contemporary video standards-based approaches where codec complexity is shared in a rigid fashion, and with the encoder bearing most of the complexity.

(2) A natural, joint source channel coding bit-stream syntax that makes the bit stream robust with respect to concerns such as drift between the encoder and decoder, transmission over loss-prone environments, and the like.

(3) High compression efficiency of the order of the current state-of-the-art video compression algorithms based on standards.

As an example, a low-encoding, high-decoding complexity configuration with efficient compression and robustness performance is of interest for an emerging class of multimedia applications that are "uplink heavy", where the video encoding device can be a processing or battery-power limited wireless device such as a handheld cell phone. Such a configuration enables light-weight, longer-lasting and less costly handheld multimedia encoding devices.

Summary of the Invention

Innovative algorithms and/or designs are included that pertain to video coding systems in general, and especially to systems based on source coding with side

information. Novel features pertain to modules in the encoder as well as the decoder, to the codec, and to systems utilizing an encoder, a decoder and/or a codec.

Brief Description of the Drawing

Fig. 1 is a schematic diagram of a source coding module for encoding with side information.

Fig. 2 is a schematic diagram of a decoder module for decoding with side information.

Fig. 3 is a schematic diagram of a codec including an encoder-decoder pair as shown in respective Fig. 1 and 2.

Fig. 4 is a graphic of an example for encoding of syndrome/intra position related mode information.

Fig. 5 is a schematic diagram of a multilevel coding system.

Fig. 6 is a graphic of an example of bit-stream syntax.

Fig. 7 is a schematic diagram of a service for multiple video streams captured from different cameras.

Detailed Description

1. Module-Level Features

1.1 The Classifier Module

A classifier module can serve to estimate the correlation distance or correlation structure between a block of data to be encoded and the predictor information that is available at the decoder. For encoding a block of data, knowledge of this distance enables the use of an appropriate family of codes for a specific situation at hand. This information is communicated to the decoder as mode information so as to enable the decoder to work with the same family of codes as was used by the encoder. If a satisfactory estimate of the correlation distance/predictor information is already known by some other means, this module can be bypassed.

As an example, owing to potential loss in the transmission channel between the encoder and the decoder, the encoder may not have knowledge of the exact predictor information available at the decoder. As another example, an encoder may not have knowledge of predictor information available at the decoder, in case the

encoder has not been able to perform a search for a predictor. Using knowledge of channel statistics, any feedback information from the decoder and the pre-compression source data correlation and the like, the encoder can draw inferences about the predictor information at the decoder and, consequently, of the innovations or new information that distinguishes a current block from the predictor information.

The well known, extensively studied compression problem, which assumes a loss-less transmission channel between the encoder and the decoder, can be viewed as a special case of this scenario. Known video compression standards for this case adopt a closed-loop deterministic approach, wherein, because of the closed-loop operation, the encoder knows exactly the nature of the predictor information available at the decoder. The encoder uses a motion search algorithm to obtain the co-ordinates/motion vector of the predictor that best matches a current block to be coded, and indicates to the decoder the predictor so chosen using motion vectors, as well as residue information/innovations between a current block of interest and the predictor information. This motion vector plus residue information forms the core of syntax for conventional video coding standards such as the MPEG-x and H.26-x series. More generally in video coding based on source coding with side information, this motion vector plus residue information that effectively determines the correlation distance can be regarded as mode information.. Since this type of mode information uniquely specifies the predictor to be used at the decoder as well as the correlation distance, the desired representation of the current block of data is uniquely determined, thus potentially obviating, or at least alleviating the need for "syndrome information" and "hash generator" fields in the bit-stream per Fig. 1. Thus, in the framework of source coding with side information, the complete syntax for the standards based methods can be subsumed in its mode information field.

Although video coding methods based on source coding with side information target the general scenario, all of the correlation distance measuring algorithms or motion search algorithms for the no-loss case that abound in the literature are relevant to the general scenario, and can be used as valuable guidelines for developing methods for estimating correlation distances. Some of them are presented below. In general motion search algorithms described in the literature, correlation estimation aims at selecting the best match in a previous frame, and then

transmitting the error between that best match and the current block. Instead, in source coding with side information, the goal is to estimate the correlation characteristics of a suitable class of blocks that will be available at the decoder. Accordingly in the following, preferred algorithms identify a block or blocks in a previous frame at the encoder, and use information derived from these identified blocks and from the current block to select the coding mode, syndrome information, intra information and hash value.

Zero Motion Residue Energy. In this method, essentially, the energy of the difference between the current block and a co-located block in the frame memory is used as a metric for estimating the extent of correlation between the current block and the predictor information at the decoder. This relatively straightforward approach has been used in a prototype system with low complexity and high robustness in the face of channel loss, but with a modest compression performance in the absence of channel loss.

Features in Addition to Zero Motion Residue Energy. The zero motion residue energy provides information about the total error between the current block and the block in the same position in the previous frame, but it does not indicate the distribution of the error energy. If, say, some of the pixels in the block are very well predicted, but others are not, then this might indicate that a displaced block will provide good quality prediction, which in turn should be taken into account in the classification process. For example, a threshold can be applied to the residue and the number of pixels above threshold counted, for use as an additional classification feature aside from the residue energy. One may also use multiple thresholds. Depending on the available encoding complexity, the residue data can also be considered in a transform domain so as to exploit the features observed in the transform domain for better classification. For example, one possibility is to apply a discrete cosine transform (DCT) on the residue data and use the relative values of the DC and AC coefficients towards the classification process. All of these methods aim at a higher system compression performance while maintaining the robustness and the low-encoding complexity properties.

Low-resolution Motion Estimation. The main advantage of using zero motion residue energy or other measurements at the zero motion location is that they

do not require any search. This ensures that the encoder complexity is kept low, though at increased uncertainty in predicting the correlation between frames and consequently lower compression performance. To reduce this uncertainty it is possible to perform reduced complexity motion estimation schemes so that better estimates are available while the encoding complexity is kept modest. Of particular interest could be low-resolution motion estimation methods for estimating motion vectors in a coarse version of the video, e.g., one where the frames are $1/16^{th}$ of the original size, and that can be used to provide improved estimates of the correlation. Further, such coarse vectors can be transmitted to the decoder as mode information without incurring any significant bit-rate overhead. The decoder in turn can use these for adapting its motion search algorithm as described below for enhanced performance. Low-resolution motion estimation can be defined with various levels of resolution, and can also be combined with techniques to measure pixel-wise rather than block-wise differences, as described above.

<u>Low-complexity Motion Estimation</u>. More generally, any low complexity motion estimation method, such as those described in the literature in contexts of video coding standards, can be used at the encoder, so as to form sharper correlation estimates and thereby to improve the system compression performance. The corresponding motion vectors can be transmitted to the decoder so that the decoder can use them to aid its motion search algorithm.

<u>Full Motion Estimation</u>. In scenarios where encoding complexity is not an issue one can perform full motion estimation at the encoder, resulting in a significantly better estimate of the correlation than the reduced complexity approaches for classification, such as those described above. A strong motivation for performing such motion estimation is to reduce the overall coding rate. Further, in the regime of transmission loss when the statistical nature of the channel impairments can be accurately established, accurate correlation estimation followed by a coding strategy appropriate for the channel at hand can result in improved end-to-end system performance. As mentioned above, the results of the full motion estimation can be transmitted to the decoder that can leverage them in its motion search algorithm.

<u>Multi-frame Motion Estimation.</u> In addition to the techniques described above, various kinds of multi-frame motion estimation can be incorporated,

feasible when multiple frames can be present in the frame memory. A principal goal of this would be increased compression efficiency.

Sub-block Motion Estimation. Within a block to be coded, separate motion estimation can be performed for various sub-blocks in order to obtain a more accurate estimate of the correlation noise. Similar to description above, the results of this motion estimation can be transmitted for use by the decoder.

Mix of Full-motion and Low-complexity Motion Estimation Techniques. The methods above have been described primarily on a per-block basis. Typically, a video frame includes several blocks. Hence, on a block-by-block basis any mix of full-motion and low-complexity motion estimation techniques can be used. For example, within a frame, full motion compensation can be performed for some subset of blocks within the frame. The motion vectors for the rest of the blocks can be estimated by interpolating the motion vectors for the subset of blocks for which full-motion compensation was performed and/or low resolution motion estimation, including zero-motion estimation, low resolution motion estimation, and the like. This allows us suitably to allocate the computational resources within a frame instead of treating all the blocks in the same manner. Similar allocation potentially can be extended across frames in order to take advantage of the temporal correlation in the motion field. For example, the motion vectors obtained from full motion estimation performed between a particular frame and the frame memory can be used as a cue for the next frame.

Classifier Operating Modes. There are two extreme cases for the classifier, which can in general operate at some intermediate point between these modes. In one case, the channel has no losses and the classifier needs to predict the correlation of the best predictor at the decoder, without being able to search for that predictor at the encoder. In this case, the encoder is limited so as to operate in a low complexity mode, and the classifier can operate by modeling the randomness of the video signal. In the other case, complexity at the encoder is not a concern, but the channel suffers from losses. In this case, the encoder can explore the set of predictors, and indeed find the best predictor. But because the channel itself has losses, the role of the classifier is to estimate the statistics of the difference between the best predictor found by the encoder and the best predictor available at the decoder. In this case the

set of possible predictors is known by the encoder and the classifier can operate by modeling the randomness of the channel and its effect on received video.

Classifier – Lossless Case. In the design of a classifier for the lossless case, the encoder can perform a search, one of those describe above, or any other motion estimation search, and can identify a best match block under the complexity constraints of the search. The goal is, given the correlation between this best match and the current block, and any additional information about the structure of the blocks, the distribution of the errors, neighboring blocks and the like, to estimate the correlation structure of the best block available at the decoder. In the lossless case, the best block at the decoder is no worse, in terms of correlation, than the best block at the encoder, as this encoder has been identified and will have been transmitted without losses. This problem can be cast as a classifier design problem, where i) relevant features are extracted from the input, such as current block and information derived by whatever search was employed, ii) a classifier is applied to each input in order to attach a class label to it, and iii) for each class a predetermined set of statistics derived from training can be employed. Such statistics can be used to estimate how much better the predictor available at the decoder will be. The design of the classifier can be performed with standard techniques after obtaining a suitable training set: candidate features can be identified, then for block and corresponding feature vector the correlation parameters of the best predictor at the encoder and the best predictor at the decoder can be generated. In some cases the two predictors will be the same. The interest here is in how well the quality of the decoder predictor can be predicted by information obtained at the encoder and summarized as a feature vector. The goals are then: i) to group together into classes those blocks that have similar characteristics in terms of decoder predictor, and ii) to identify those features that provide the best separation between classes. The first goal can be accomplished by using the differences between encoder predictor and decoder predictor to establish an "ideal" class. Since this ideal class is based on knowing information available only at the decoder, it is not available in practice. Thus, the next step is to design a classifier that operates only on the features available at the encoder and that best approximates the ideal class. Algorithms to accomplish this have been developed in the context of designing quantizers optimized for classification. The second goal, the selection of

best features, can be accomplished by measuring, e.g. using mutual information, how much each individual feature in the feature vector is able to predict the final outcome. This information can then be used to use weighted distances, e.g. the Mahalanobis distance, in the feature space and even to discard some of the initial features if they do not contribute significantly to identify the right class. The process described above is optimized for the given training set. If the training set is representative of the video sequences to be encoded, then a fixed classifier can be used, otherwise adaptive techniques may have to be employed, for which the decoder feedback would be needed.

Classifier – Lossy Case. As discussed, channel losses introduce additional uncertainty because even if the encoder searches for the best predictor in previous frames, there is no guarantee that this predictor will be available at the decoder. In many cases of interest the encoder can be limited in complexity so that the best predictor at the encoder will not be known, and there can also be losses, so that both types of uncertainty, source and channel, will be combined. In the case where the encoder is not complexity constrained, the encoder can identify the best blocks among the set of candidate predictors. For each of these best predictors the encoder will have a measurement of correlation. Also, the decoder can have a measurement of the distance between these predictors in the previous frame, e.g. the distance between the top right pixel of each of these blocks. The distance between predictors can be used to evaluate the likelihood of all the predictors being simultaneously lost or affected by losses in previous frames. A possible strategy for this situation, then, is to identify a set of "good" predictors that have sufficient distance between them to make it unlikely they will all be lost. Then the parameters of the encoded data are chosen so that it can be decoded when the side information at the decoder is any of the predictors in the selected subset.

Role of Feedback. The presence of a feedback path from the decoder to the encoder can greatly improve the overall system performance with respect to the three metrics of compression, complexity and robustness. Depending on the nature of the information, feedback can be useful for the encoder classifier module for improving its classification process and refining the classification functions that are being used by taking input from the decoder. This will result in enhanced

compression efficiency. Further, feedback information can also be useful for decreasing complexity of the encoder classification process. For example, if the decoder can send as feed back, the motion vectors produced by it during the motion search process, the encoder can use them to estimate the nature of motion in the video for future frames and hence the correlation estimates. Feedback information can also prove to be useful for inferring the nature of the channel at hand and can be used by the encoder to choose the appropriate encoding parameters so as to achieve robustness in transmission.

1.2. Mode Information

As described above, the classifier module estimates the correlation distance between a current block of data and the predictor information present at the decoder. These estimates are then communicated to the decoder through the mode information field in the bit-stream syntax for video coding systems based on source coding with side information. As an example, motion vector information and residue information which form the main part of known video compression standards such as MPEG-x and H.26-x can be used to provide an estimate of correlation distance. Under this notion, the mode information field subsumes the entire syntax associated with the standards-based approach to the pure compression problem, as the objective of standards-based syntax is essentially to communicate the correlation distance information to the decoder.

Typically, mode information maintains delineation in the bit-stream enabling synchronization between the encoder and the decoder and can help to improve the decoding performance by aiding the decoder. For example, as noted above, depending on the classification algorithm deployed, any cues such as motion vectors observed by the encoder can be transmitted to the decoder to enhance decoder performance. Such information is one instance of many types of information that can be transmitted as mode information to the decoder.

As another example, when the classifier operates on a dataset/block that is a collection of the data/coefficients that are actually encoded, mode information compactly represents the nature of correlation experienced by various coefficients. In appreciating the value of such mode information, it is noted that significant improvements in compression performance can be achieved by

recognizing that decisions about (i) whether or not to use syndromes for encoding, and (ii) how many bits to encode using syndromes, should ideally be made at the coefficient level. However, if different decisions are made for each coefficient of each block, this will result in a significant increase in the overhead to be sent to the decoder. Thus, an important aspect of system design is to maintain maximal flexibility in mode decisions, while not requiring excessive overhead. Some mechanisms to ensure that the mode information does not take excessive bit-rate while maintaining good flexibility are:

"Intra" Line. In every block of data, the first fraction of transform coefficients is usually highly correlated with their counterparts from the predictor block while the rest of the coefficients typically are weakly correlated. Hence, some performance improvements can be obtained by treating these two sets of coefficients differently, for example, by syndrome encoding the first fraction of the coefficients, and intra-encoding the remainder. The mode choice can reflect the position of the intra line beyond which all the coefficients are intra-encoded. This approach has been implemented in the prototype system.

Mixed Section and Intra Line. A ready improvement provides for some coefficients in the syndrome-coded section to be intra-coded, with a block divided into two parts, namely a mixed section consisting of syndrome and intra-coded coefficients and an intra-coded section. A convenient way to indicate to the decoder the boundary between mixed and intra-coded coefficients is to transmit as overhead the position of the last coefficient that is sent in the mixed section. To reduce the overhead, the set of admissible positions for such a boundary can be restricted.

Efficient Run-length Encoding of Modes. The mode information for each coefficient in the mixed section can be Intra/Syndrome-Class. The Intra field indicates whether or not the coefficient is intra-coded. Under the bit plane representation of symbols, if a coefficient is syndrome encoded, the Syndrome-Class can indicate the number of bit planes of the coefficient that are encoded via syndrome– the higher the correlation, the greater the number of bits that can be encoded via syndrome– and hence also the number of bit planes that can be encoded as refinement bits. Suitable codes such as run-length entropy codes can be used to

support alternating representations (from syndrome to intra) for coefficients, and where different syndrome-classes are possible. Such a representation can include for each block: header information containing the position of the intra line, run-length encoding of the mode information for all those coefficients sent in mixed section, standard representation for all the intra coefficients (e.g., like H.263+ intra-coding), syndrome bits, and refinement bits.

Focus in the following is on run-length encoding of mode information. The mode information should tell the decoder the number of bits that can be predicted from the frame memory, or at least an estimate of this number. Potentially this can vary for each coefficient. Further, as mentioned above, the syntax should be sufficiently flexible to let INTRA coefficients into the SYNDROME coded zone. On this basis, when t consecutive coefficients have the same number m of predictable bits, we speak of a run of m and run length t. To take advantage of such runs, a run-Length code can be used. In order to allow INTRA coefficients into the SYNDROME zone, a run-Length code can have a special symbol to identify INTRA coefficients. Thus, a run-length is represented by a tuple (LEVEL, RUN) where LEVEL represents the number of predictable bits, with a special symbol for INTRA coefficients, and RUN represents the length of the run. Another symbol, EOB (End Of Block) serves to tell the decoder where the SYNDROME zone ends. INTRA coefficients within the SYNDROME zone can be packed together with the INTRA part of the bit stream, and coded using a regular Run-Length code. The following is illustrative:

Example. For Fig. 4, these values represent the number of predictable bits for each coefficient before the intra-line:

S	S	S	S	S	I	I	S	S	S	S	S	S	S	S	S
2	2	2	1	1	0	0	2	2	1	2	2	2	1	1	1

These values are coded as (3,2) (2,1) (2,I) (2,2) (1,1) (3,2) (3,1). An INTRA coefficient should be inserted within the SYNDROME zone when the coefficient value is 0. In this case the Run-Length code is used advantageously to code the INTRA coefficients. An INTRA coefficient should be inserted also when

the coefficient is not strongly correlated. In this case, sending the refinement bits possibly can be more expensive than just sending the entropy-coded INTRA value. This is an example of the type of decisions to be made by the rate controller.

Role of Feedback. Specific error concealment technique used at the decoder and, possibly through decoder feedback, knowledge of the prevalent channel conditions can be used to modify the mode information decisions. Based on the error concealment technique and the channel loss probability, the encoder can choose the position of the "intra" line, the coefficients to be syndrome encoded, the number of bit planes to syndrome encode, and the specific channel codes to use in the syndrome encoding process in a rate-distortion optimal sense. In particular, knowledge of which packets were lost in previous frames along with the characteristics of the error concealment will allow to estimate the additional noise affecting predictors at the decoder. This knowledge can be incorporated into the mode selection. In particular, as in the well-known ROPE algorithm, an Intra block may have to be sent if past errors are expected to add significant noise.

1.3 Syndrome Encoding

In some scenarios, the nature of classification information is such that it does not uniquely determine the desired representation of the current block of data. In such a case, additional information in the form of syndrome information about the current block of data can be transmitted to the decoder. The decoder can use the mode information and the syndrome information together to decipher the desired representation of the current block of data. In the framework of linear channel codes for generating syndrome information, the syndrome bits effectively index a collection of uncertainty lists where each list contains a collection of candidate representations for the current block of data. Typically, the amount of syndrome information generated can depend upon the nature of mode information. For example, when the mode information can provide only a weak inference about the current block of data, more syndrome information might be required.

Also, as noted above, in some cases the operational dataset/block can consist of a collection of information data/coefficients that are actually encoded. In such cases, typically, some of the coefficients in a block could be encoded through syndrome encoding. Given the bit plane view of a coefficient where different bit

planes have different amounts of predictability from the predictor information, a framework for syndrome encoding based on multi-level codes is suitable where separate channel codes can be used over different bit planes.

Multi-level Coding Framework. In this framework, bits at any bit plane can be sent un-coded as well. The channel codes used at each bit plane can be varied for every block. The decoder can be informed of the choice of channel codes used through the mode information. Details of a possible implementation of a multi-level coding algorithm are as follows:

The input to the multi-level coding algorithm can be the target distortion, the target probability of decoding failure P_e, and the correlation noise statistics. The target distortion can be mapped to a target step-size δ. Given these three inputs, the multi-level coding algorithm can compute the "un-coded probability of error", or $P_{e,uncoded}$ which is the probability of error before error-correcting codes are applied. From $P_{e,uncoded}$ and the correlation noise statistics we can find the target step-size Δ that will result in a probability of decoding error $P_{e,uncoded}$. If Δ is greater than target step-size δ, then multiple levels will be required to refine the coarse step size Δ to the target step-size δ. The number of levels required will depend on the quantization strategy. For one-dimensional scalar quantization, the number of levels required will be $\log_2(\Delta/\delta)$.

Fig. 5 shows the multi-level coding framework using one-dimensional scalar quantization, with three levels required. For this example, the source X is quantized on the quantizer which has the points "001" and "101". This quantizer is indexed by the two lowest significant bits of X, which are 0 and 1. In an un-coded transmission strategy, the encoder communicates to the decoder that the two lowest significant bits of X are 0 and 1, which also are called the refinement bits. The resulting probability of error is $P_{e,uncoded}$. To drive this probability of error down to the target probability of decoding failure P_e, one of two strategies can be used, namely an un-coded strategy or a coded strategy. An un-coded strategy would specify the most significant bit of X. This strategy would require one extra bit per coefficient. On the other hand, the coded strategy would drive the probability of error down to the target P_e through the use of appropriate channel codes. For example, if

the total number of coefficients is n, then the number of coefficients in error is approximately n times P_e. If an (n,k_3) channel code can correct n times P_e errors, then the number of extra bits that need to be sent per coefficient is $(n-k_3)/n$ which is less than 1, the number of extra bits that need to be sent per coefficient for the un-coded case. Codes can also be applied at higher levels in a similar manner, as shown in Fig. 5.

Role of Feedback. When possible, decoder feedback can be used to unburden the encoder classification module substantially, and hence to reduce encoder complexity. As noted above, the role of the classification module is to estimate the correlation between the current block of data and the data in the decoder frame memory. The tighter the estimate of the correlation, the better the overall system compression performance. Using a framework such as the multilevel codes framework, the encoder can start off by assuming tight correlation and thus send fewer syndrome bits to the decoder. The decoder attempts to decode with the available information and informs the encoder if it fails. The encoder can then send more bits such as by going deeper into the multi-level coding tree. This process can be continued until the decoder decodes correctly. In such a setting it is possible that the encoder merely generates syndrome information bypassing the classification module and thus minimizing the encoding complexity. By repeated or limited feedback to the encoder, the decoder draws just enough syndrome information from the encoder to correctly infer the desired representation of the current block of data. Thus, the system can attain best possible compression performance with low encoding complexity.

1.4. Hash Generation

The availability of mode information/syndrome code may result in an ambiguity at the decoder as to the desired encoder codeword. The decoder can try one or more candidate predictors to infer the correct codeword from the ambiguity set. Depending upon the procedure used, it is possible that the candidate predictor(s) can decode some codeword(s) from the ambiguity list. An extra mechanism can be deployed by the decoder to operate on this set containing decoded codeword(s) in order to uniquely determine which, if any, codeword from this set is the desired encoder codeword. Such a mechanism can reliably authenticate the validity of the

decoded codeword with respect to the desired encoder codeword. As an example, the use of a hash function that generates a signature of the block data can suffice for this purpose. The hash signature can be transmitted to the decoder. The decoder can generate the hash signature(s) for decoded codeword(s) and compare it with the received signature when authenticating the decoded codeword(s). There are many possibilities for choosing good hash functions, of which some are listed below.

Cyclic Redundancy Check (CRC). A CRC checksum generated on the binary representation of the quantized block data prior to the syndrome encoding of the data can serve as a hash codeword. At the decoder each decoded codeword can be used to generate a checksum with the same CRC function, and if the hashes match, the decoding can be declared to be successful. Such an approach has been used in the prototype system.

Soft Hash. There is a concern with the above approach in that it does not provide an indication as to how close a given decoded codeword is to the original block. This concern can be addressed by soft hash, for example, by combining the intra information transmitted for a given block with a shorter CRC code, which in some cases may be completely skipped. Thus, the intra information is first used to eliminate many potential candidates, and then, when the syndrome information is used to decode based on each of the remaining candidates, the resulting decoded block is tested based on the CRC code. Such an approach can also be useful for directing the choice of the future candidate predictors based on the outcome of the decoding with the current and previous candidate predictors. As an example, if after decoding with a candidate predictor the decoded codeword is not sufficiently close to the original block, the decoder can use this information to eliminate future candidate predictors that are similar to the current candidate predictor. Examples of soft hash information are included below.

MSB Intra Coding. As one possible way of generating intra information, the most significant bit plane (MSB) of selected coefficients can be sent as a part of the hash codeword, before using syndromes to encode lower significant bit planes. Then the search algorithm at the decoder can eliminate blocks from the search that do not have the desired MSB signature.

Available Intra Information. The decoder can also use some of the information transmitted in the block bit-stream such as mode information as well as intra-encoded coefficients to reduce its search space and remove candidate predictors that are not consistent with this information. Another example of available intra information can be parts of syndrome information that index uncertainty lists of size one, i.e., they have no uncertainty. As an example, sometimes refinement bits for a coefficient can be sent as a part of syndrome information.

Shared Hash Tables. Intra information can be used to complement the hash information and, conversely, the hash information in fact can provide some Intra information as well. As an example, assuming that an 8-bit CRC is used, any input block or portion thereof is mapped into one of 256 different classes. Multiple input blocks belong to each of these classes, and these blocks will have very different characteristics. However, as the CRC is sent to the decoder, a mechanism is desired for exploiting this information for coding purposes. One such mechanism is to identify a relevant characteristic of the block, e.g. the position of intra zeros before the intra-line, and then to create a table such that for each of the 256 classes, this information is tabulated for the most likely blocks in the class. Thus, the encoder can compute the hash value, and then can use overhead bits to communicate to the decoder as to which, among the blocks in the class, is the one being transmitted, if it is one of the most popular ones. Otherwise the hash is used in the standard way.

Continuous Error Detection (CED). CED is often used as a replacement for CRC in communication scenarios. The CED decoder processes each bit as it arrives, and each bit can potentially indicate that an error has occurred somewhere in the transmission, though it cannot pinpoint the location of the error. Such a CED decoder need not wait for the entire block to be received before it can detect transmission errors, unlike CRC. Similarly, the CED can be used as an alternative to CRC, in which CED will check whether the decoded block is the same as the original block. Advantageously here, the decoding process can be terminated as soon as the CED decoder detects an error. By not having to decode the entire block, complexity at the decoder is reduced at the price of CED being inferior in terms of coding efficiency. Use of CED entails a tradeoff of coding efficiency for complexity savings at the decoder.

1.5. Bit-stream Syntax

In view of the above, the following can represent the bit-stream syntax for a given block. For each block, the bit stream can include the mode information, syndrome information and hash information fields. In certain cases, some of these fields can be omitted. For the case of mode information of the type of "mixed section/intra line" and "efficient run-length coding of modes" per Section 1.2 for example, we could have for example, the bit-stream syntax can be as shown in Fig. 6:

QUANTIZER, quantization step size,

MODE INFORMATION, as described above,

INTRA BITS, entropy coded using Run-Length codes. The INTRA coefficients inside the syndrome zone and the regular INTRA coefficients can be coded together.

SYNDROME BITS

HASH BITS

REFINEMENT BITS

1.6. Decoder Motion Search

As noted in Section 1.4, the decoder can try one or more candidate predictors to infer the desired encoder codeword from the ambiguity set. The decoder can generate the candidate predictor(s) by using a search procedure. The search procedure can be similar to the motion search procedure used for motion estimation at the encoder in the context of standards-based video compression. Thus, all the advances in motion estimation algorithms that have been made in the context of conventional standards can be leveraged at the decoder here, so as to give the best performance with as little complexity as possible. Further, the mode information generated by the encoder can also be used to influence the search procedure. Examples of various techniques that can be deployed at the decoder include:

Information Available through the Bit Stream. The decoder uses the intra information already received for a given block to find suitable candidates for best match in the previous frame. Depending upon the type of mode information generated at the encoder, the following information could be available at the decoder: all coefficients beyond the intra line coded in intra mode, the position of coefficients before the intra line that are zero, the sign of those coefficients that are not zero

before the intra line, the percentage of energy before the intra line, the encoder motion vectors for the given block, and the like. Further, the use of soft hash mechanisms as described above can enable the decoder to prune its search, thereby reducing the complexity.

Fast Metric Computations. Typically, a codeword in the ambiguity list is inferred to be correctly decoded by a particular predictor if the codeword is the best according to a particular metric. By developing fast methods for metric computations, we can potentially speed up the decoding process. The standard motion estimation system takes two blocks, namely the original and the candidate block in the frame memory, and computes a suitable difference metric, e.g. sum absolute difference (SAD) or sum squared difference (SSD). It is well known that some of these metrics can be computed exactly or approximately in the transform domain. In the present case not all the coefficients will be available. For some coefficients only the sign or the significance are known. Several methods are available to approximate a metric, e.g. SSD, with partial information. For example, only the information of coefficients beyond the Intra line may be used. Or those coefficients can be used first, to select a few candidates, and then the ties can be broken based on the Hamming distance between the significance of the coefficients before the Intra line. Or one can approximate the coefficients before the intra line using their sign and the average energy. Generally speaking those blocks having energy in the high frequency will be suitable for motion estimation at the decoder, while those that are primarily low frequency will be difficult to estimate. Thus, an encoder may choose to use motion estimation at the encoder for those blocks for which motion estimation is unlikely to work at the decoder.

2. Codec-level Features

Codec-level features can be incorporated in systems based on source coding with side information. Examples of such features include rate control, scalability, and the like.

2.1. Rate Control

For efficiency in a video coding system, a rate controller can match the mode of operation of a codec to a desired bit-rate. A typical rate control algorithm decides how many bits to use per block or frame, and what coding mode to use for

each block or frame. In a video coding system based on coding with side information, the role of the rate controller corresponds to that in a conventional codec, i.e. making mode and coding decisions to achieve the minimum distortion for a desired rate. It is useful to view the rate control algorithm as a two stage procedure. First, the target video frame can be encoded in intra mode. This entails selecting quantization parameters for subunits of the frames, such as blocks or slices, in order to achieve a desired overall rate. Second, the encoder can select the information to be sent in order to reconstruct each of these Intra coded blocks at the decoder. For example, the encoder may choose to send the coded blocks directly using Intra mode. Or the encoder may choose to skip a given block, because the error with respect to the block in the same position in the previous frame is small. Or finally, the encoder can choose to transmit some of the coefficients of the block using syndromes, and others in intra mode, using methods as those discussed above. The rate control algorithm aims at sending the minimum number of bits to the decoder that can provide a given target quality at the decoder or, conversely, to achieve the best possible quality while consuming a pre-specified number of bits. The parameters that can be chosen include the initial Intra mode quantization which can vary from block to block, the mode selected for each block, i.e. Intra, Skip or Syndrome, and, in the case of Syndrome mode, the position of the intra line, the number of bits of refinement and the like. The rate control problem differs significantly from that encountered in standard video codecs. In particular, while in a standard codec the encoder has exact knowledge of the quality achievable at the decoder given a choice of operating mode (e.g., quantization choice for a block), this is no longer true in the types of compression systems under consideration. In these systems, there is uncertainty about what the decoder can produce because the encoder does not have exact knowledge of the predictors the decoder can have access to. As an example, after performing a suitable classification, the encoder can have an estimate of how good of a predictor can be found at the decoder. Roughly speaking, the better this predictor at the decoder is, i.e. the more correlated it is with the input being encoded, the fewer bits will be needed to reconstruct the current block correctly. Thus, the encoder can reduce the number of bits it spends to represent a block at the expense of increasing the risk of incorrect decoding in case a suitable predictor cannot be found at the decoder. In this situation,

the quality metric to be used by the encoder becomes probabilistic, rather than deterministic as in the standard video coding case. A basic tool for rate control to be used in a preferred system is to compare the rate required for intra coding to that required for syndrome based coding, and choose the mode that yields the lowest rate among these two. Assuming correct decoding, both modes result in the same decoded block. This decision can be augmented by a rate-distortion based decision. In this case the encoder computes the rate and distortion associated to each of the coding choices. The main difficulty in doing this is that the distortion is not known exactly when a syndrome mode is used. This can be solved by computing the distortion of each syndrome mode as the sum of distortions in each possible decoding scenarios, weighted by their respective probabilities. For example, assuming a correct and an incorrect decoding scenario, in the correct decoding scenario the distortion will be the same as that achievable in intra mode. If correct decoding cannot be achieved, then the distortion with respect to the collocated block in the previous frame could be used as an estimate. This distortion would be weighted by the estimated probability of error. The rate and distortion can then be combined into a single Lagrangian cost function using standard techniques. This technique will tend to choose syndrome mode, intuitively, when a combination of the following factors is favorable: i) syndrome mode represents a significant reduction in rate with respect to Intra mode, ii) the probability of error is very low, and/or iii) the distortion with respect to the collocated block in the previous frame is relatively small. In some cases the technique can be complemented or replaced by a metric that directly takes into account the probability of error.

<u>Constraint-driven Rate Control</u>. For transform coefficients, blocks, frames, and the like, a general rate controller can take into account factors such as source characteristics, target bit-rate/quality, and target complexity. An algorithm for rate control can be based on making decisions that are best in terms of given overall rate and given complexity. For example, based on a given decoding complexity, the encoder can choose to encode in Intra mode, which is a low complexity mode, a predetermined number of blocks, and then to optimize this decision. The number of intra blocks can be based on available complexity considerations, while the choice of which blocks to send in intra mode can be driven by the rate and distortion

characteristics of all the blocks. Thus, the blocks for which the syndrome encoding provides the least gain in coding performance should be considered prime candidates for intra coding. Sending blocks to the decoder in Intra mode allows the decoder to perform motion estimation on those blocks, which in turn will help initializing the search for neighboring blocks that have been sent in Syndrome mode.

Bit Stream Field Skipping / Information Reduction. Depending upon the situation and the available bit-rate, some parts of the bit stream can be skipped, or the information transmitted in certain fields can be reduced. As an example, in some cases blocks that are not sent in Intra mode contain intra-coded information. As noted in Section 2.1.6, even partial intra information can lead to acceptable motion estimation at the receiver. The encoder can take this into account when deciding how to encode a given block. For example, if a block is recognized by the encoder to be such that motion estimation at the decoder is potentially successful, then the encoder can decide not to use a hash codeword for that particular block. Conversely, if the encoder estimates that accurate motion cannot be found at the decoder given the block structure, it can choose to use a hash codeword, or to perform low complexity motion estimation, and send a reduced size hash codeword. Likewise, it is desirable to have a rich mix of motion modes with, for example, full-motion search being advisable on a small percentage of the blocks, thereby providing an "anchor" or "beacon" for neighboring blocks. Coarse-motion estimation can be advisable for a small subset of blocks, zero-motion for yet another subset, and no-motion (i.e. Intra-mode) for others. Being able dynamically to classify these blocks into different classes can significantly enhance the performance for a specified bit-rate.

Role of Feedback. Feedback from a decoder can also assist the rate control mechanism. For example, for transmission over a loss-prone channel, decoder feedback can provide an estimate of channel statistics to the encoder. The encoder can adjust the quantization step size and the amount of syndrome information based on the feedback so as to enable the decoder to function well. When the channel has a high loss, the encoder can increase the amount of syndrome information. More information can help the decoder by reducing its search space.

2.2 Scalability . A major challenge in video coding has been the design of an efficient coding format that provides scalability, useful to ensure robust video

communications over unreliable channels of various types. A preferred codec can support scalable encoding including spatial, temporal and SNR scalability. The scalable bit stream can consist of a base layer encoding and multiple enhancement layers. For decoding each layer, the decoder can use all past decoded frames as well as the layers already decoded of the current frame. The layers may be encoded using a standards compatible framework or the preferred side-information based codec. Such a method is naturally more robust to channel errors as compared to standard coders due to the inherent robustness of the side-information based coding scheme. For a side-information based scalable coding scheme, the enhancement layer can still be decoded, even when parts of the base layer are not received correctly due to channel errors, if the codes used for the enhancement layer are strong enough. This is not generally possible for a standard codec.

A prediction based coder needs to keep multiple predictor copies at the encoder since different decoders can have different decoded versions based on how many layers are decoded. On the other hand, the scalable side-information based video coder only needs to keep an estimate of the correlation between the current frame and the different possible decoded versions at different decoders. This allows the side-information based coder to scale to a large number of possible encoding rates. One algorithm for video scalability that has received some attention in recent years is known as Fine Grain Scalability (FGS) within the context of the MPEG-4 standard. In this algorithm, the base layer is encoded as a standard MPEG stream. Then each of the enhancement layers is encoded by computing the difference between the decoded base layer frame and the original, and successive bit planes of information of this error frame are transmitted.

Fine-Grain Scalability. In a preferred system, the motion vectors are assumed to have been received correctly, in the base layer, and therefore a correspondence can be established between blocks in a current frame and blocks in the previous frame. Assuming that the previous frame has been received, one objective of scalability can be to decode blocks in the current frame with a resolution that matches what was received in the previous frame. For a syndrome based encoder, this can be achieved by (i) estimating the number of coefficient bit planes that can be reliably obtained from the previous frame, (ii) copying only the reliable

bit planes, and (iii) sending multiple syndrome codes corresponding to the different decoding scenarios. When only one block is used for decoding, namely the one pointed at by the motion vector of the base layer, it is not necessary to include CRCs in the bitstream. Reliable decoding at the appropriate resolution can be achieved by a correct estimate in step (i) and by sending in step (ii) codes that can correct errors in various bit planes.

Standards-compatible Base Layer. As described above, one aspect of a preferred algorithm lies in the ability to mix Intra coded and Syndrome coded coefficients. By appropriate selection of the coefficients sent in each mode it is possible to provide a standard compatible substream that is amenable to fast decoding. For example, by sending the DC value of each block in Intra mode, an image of a fraction of the original image's size can be decoded without substantial decoding complexity. This can serve, for example, in the context of a digital camera, as a coarse version of the captured video scene to be used in a viewfinder or to provide a preview of the encoded video.

As another example, a standards-compatible base layer can be encoded with MPEG, and the next enhancement layer encoded with side-information principles. The enhancement layer decoder can use the MPEG decoding of the base layer as a possible side-information for decoding the enhancement layer.

3. System-level Features

System-level aspects can be used to enhance a video coding system based on source coding with side information.

3.1 Trans-coding

Trans-coding of the syntax associated with a video coding system based on source coding with side information to/from a standards-based format is desirable for increasing inter-operability. There can be useful applications in which the trans-coding algorithms are applied with the starting point being an already encoded stream. For example, a stream in a given format, e.g., motion-JPEG or MPEG, is taken as an input, and a bit stream in format of source coding with side information is generated utilizing readily available information.

Trans-coding from Motion-JPEG. When considering motion-JPEG as the starting point, the goal of trans-coding can be to significantly reduce the overall

rate needed for the sequence by exploiting redundancy between frames. In this case, the trans-coder simply needs to decode the JPEG entropy codes of each frame, and then decide, based on estimated correlation with the previous frame, which of the quantized coefficients should be sent as is, and for which syndrome encoding could be used. As no transform is used, processing can be very fast. A particular scenario of interest for this trans-coding application is one where a digital camera produces motion JPEG encoded streams. The trans-coder can then operate online or offline to reduce the storage or communication requirements with minimum complexity.

Trans-coding from MPEG. For an MPEG encoded format, an advantage of a syntax based on source coding with side information can be to enhance robustness over the MPEG stream, and make it possible to decode even in the presence of packet losses. In this case, the trans-coder performs a partial decoding of the MPEG stream. For example, I-frames and Intra blocks in other frames can be left as is. Inter and bi-directionally predicted blocks could be decoded, and then re-encoded in a way such that some of the corresponding coefficients are coded in Intra mode. In this case the correlation is known. assuming that the motion vector has been chosen to minimize the distortion between current block and blocks in the previous frame. In a suitable design, the motion vectors are transmitted after trans-coding, and each block is represented in a mixed Intra and Syndrome mode. At the decoder, in the absence of channel errors, the algorithm can decode blocks using the corresponding motion vectors along with the corresponding block in the previous frame. If an error has occurred, the decoder can search for alternative blocks that would enable decoding by using either a CRC sent by the encoder or the Intra information included in the received block.

Trans-coding to Motion-JPEG/MPEG. Methods for such a conversion have been described in the above-referenced U.S. Patent Application No. 10/651,854 of August 29, 2003. The block-based approach for source coding with side information can be of particular value here, as both motion-JPEG and MPEG have block-based architectures.

Trans-coding to/from an authenticable format. As described above, the use of hash information can enable to reliably authenticate the validity of a decoded codeword with respect to the desired encoder codeword. By inserting/removing hash

signatures for the encoded codewords, one can trans-code to/from an authenticable format.

3.2 Side-stream Enhancement to Standards-based Codecs

A video coding system based on source coding with side information can be designed to co-exist with a conventional standards-based video coding system, thereby offering significant advantages.

One concern in prediction-based codecs, e.g. MPEG and H.26x, is with so-called drift. Predictive-encoded bit streams that are fragile under losses. In predictive video coding, only the difference between the current frame and the previous frame(s) is sent to the decoder. If for some reason, e.g. due to channel errors, the previous frame(s) are not available at the decoder, the reconstruction of the current frame at the decoder will be inaccurate. This error can propagate, as the next frame will be predicted based on the current frame, and the encoder is unaware of the fact that the reconstruction of the current frame at the decoder was incorrect.

A coding system based on side information does not suffer from the problem of drift because in principle the current frame is not encoded based on any particular predictor. Such a bit stream can be decoded as long as there is some predictor available at the decoder within the correlation distance used for encoding.

The error resilience of video coding based on side information can be used to add robustness to a conventional MPEG or H.26x bit-stream. An augmentation to conventional video coders (like MPEG and H.26x) can be made in which a side stream is sent along with the conventional MPEG or H.26x encoded bit stream. The side-stream data can be based on source coding with side information, and can be used to correct "drift" errors at the decoder. An instance of the details that can be involved in constructing such a system follows below.

In conventional video codecs each block within a frame is coded as either "intra" or "Inter". Given the channel model, it is possible to estimate the "expected distortion" for the block at the decoder for each possible coding mode, intra or Inter. The decision to code the block as intra or Inter can then be made based on which of the coding modes is better with respect to Rate/Distortion.

The side-stream technology can be used to increase the number of coding modes. For the Inter mode there are choices as follows:

1. Block reaches decoder leading to a distortion D_1.

2. Block does not reach the decoder, and the decoder has to resort to error concealment using surrounding blocks leading to distortion D_2.

3. Block does not reach the decoder, and no adjacent blocks are available, leading to distortion D_3.

A target distortion D can be sought as less than D_1, D_2 or D_3, resulting in corresponding coding modes augmenting the Inter coding mode: For target distortion D_1, the side-stream is coded assuming that the distortion between the decoded data and the original data is D_1. The side-stream is used to refine the decoded data to a distortion D. D_1 need not be equal to the target distortion D even though the block has reached the decoder correctly, due to previous errors in transmission. For target distortion D_2 or D_3, the side stream is used to refine the decoded data to a distortion D.

There result five coding modes, namely the three side-stream cases listed above and the conventional Intra and Inter, with each leading to different expected distortions. The coding decision can be based on which of the coding modes is preferable with respect to Rate/Distortion. For the side-stream cases with target distortions D_2 and D_3 it is unnecessary to even send the "Inter" encoded block, as the side stream sends enough data to refine, to the target distortion D, data already present at the encoder. For these cases the coder defaults to a side-information based video coder.

With target distortion D_1 for the side-stream option, the resulting encoder will be fully compatible with a standard conventional coder such as MPEG or H.26x. A decoder that does not include the side-stream option will still be able to decode the bit stream, as there is no change to any part of the standard Inter-coded bit stream. A decoder with side-stream functionality will be able to achieve lesser distortion using the side-stream option.

3.3 <u>Feedback</u>

As noted above, feedback can enhance the performance of a video coding system based on source coding with side information with respect to all the 3 metrics of codec complexity, robustness and compression.

3.4 <u>Multicast Application</u>

In multicast applications, multiple receivers can simultaneously receive the same multimedia stream. Depending on their reception profile, as some receivers have more loss-prone channels than others, various receivers can have different contents in their frame memory. Source coding with side information can be used to attend to these various receivers simultaneously. For example, when multi-level codes are used, the encoder can transmit the various syndrome bits, on different multicast channels. Each receiver can subscribe to a different number of channels, depending upon how many bits are needed by them to decode correctly.

Another example is in scalability in the multicast setup. The encoder can output a base layer stream and a number of enhancement layer streams on different multicast groups. Depending upon the rate available, each decoder will subscribe to a certain number of multicast groups. The codes to use for each enhancement layer will depend upon the side-information available to the typical receiver subscribed to the multicast group containing that enhancement layer. As in Section 2.2, the encoder need not keep multiple predictor copies. It only needs to keep track of the statistical correlation between the current frame and the different qualities of the side information present at the various decoders. This allows the scheme to scale to a large number of rates. The quality of the side information present at a particular decoder is affected by the number of multicast groups it has subscribed to as well as the channel between the encoder and the decoder.

3.5 Multi-source Application

In a multi-camera environment, preferred techniques can be used in distributed compression of multiple video streams captured from different cameras that can be wirelessly networked as illustrated by Fig. 7, for example. Where communication for joint encoding between the video sensors may not be practicable, e.g. due to bandwidth and power constraints, predictive encoding would be unable to utilize any correlation among the video sequences. However, in a system with multiple video sensors, one of the sensors can send its data to the central unit without regard to the other sensor, thereby enabling the latter to encode its data with regard to the side-information data from the first sensor at the central decoding unit. All relevant spatio-temporal correlation between multiple video streams can be utilized in

a fully distributed way, resulting in enhanced compression/power efficiency over conventional techniques.

Another challenge in such a scenario includes integrating the individual resolutions of these cameras to effect a virtual "super-resolution" camera that can be accomplished through sophisticated back-end processing at the base-station end of the wireless network. While computer vision techniques are available to do multi-view fusion of the outputs of individual cameras, what makes the present scenario unique is the need for these to live in a bandwidth-constrained environment, e.g. due to the wireless channel, that can dictate the need for distributed compression. Yet another challenge can be distributed co-ordination of cameras to accomplish features such as pan, tilt, zoom and the like.

The decoding unit at the back end or base station or at any server connected to the base station can perform all the demanding processing tasks. As an example, the decoding unit can perform multi-camera classification for distributed video decoding from the multitude of cameras observing highly correlated scene information. It can also perform multi-view fusion of the output of the individual cameras using multi-view computer vision techniques, for example, adapted to the distributed compression environment.

The broadcast nature of the wireless network environment can be exploited to benefit the system. For example, the decoder can broadcast of complex tasks such as motion vectors of past frames for use by the individual camera encoders. The decoder can also broadcast of the classification parameters / classification modes to the individual cameras over the wireless downlink. The decoder can also broadcast of packet loss information for use by the individual camera encoders to dynamically change their classification modes, as well as to repair the effects of past losses, e.g. through re-sending of past data that was corrupted in an ARQ-fashion, or to send "incremental syndromes" of past data to allow for decoding correctly in the retransmission phase. The decoder can enable adaptively changing the instantaneous complexity among the encoder units and the decoding unit to effect dynamic load balancing depending on individual load dynamics. Individual cameras can also broadcast the results of their processing, such as motion vector search results, over the wireless network for possible use by other encoding units to reduce

their complexity. By encrypting this data, privacy can be preserved as well. The decoder can allow for the dynamic clustering of network cameras to accomplish virtual zoom/tilt/pan features and the like.

Computer vision pre-processing techniques can be combined with the framework of distributed video coding to enhance the overall system performance. As an example, in a situation where a single camera is used and some preprocessing is performed before encoding to determine whether some information worth transmitting is being captured, preprocessing tasks can include motion detection, possibly after compensating for known camera motions, for motion tracking of an object in the scene. If after such processing no motion is detected, a very limited amount of information can be sent. In some cases, the processing can be done on the source side, prior to encoding. Or, with feedback from the decoder to the encoder, information can be communicated by the decoder based on available side information. With this information, the encoder can generate estimates of correlation at the encoder in different ways depending on whether a particular block is in a region where motion has been detected or not. If motion has been detected, then the encoder will use the known motion instead of, say, the zero motion, to estimate the correlation for that block. As mentioned above, in case of multiple cameras any such information inferred by the decoder can be broadcast for the benefit of all encoders.

Distributed coding techniques can be used also for the case of compression of sensor data captured at high resolution from multiple sensors. One of the major barriers for the deployment of very high definition solid-state video cameras is the bandwidth required to read pixel data captured by each of the multiple sensors so that it can be compressed and stored. More specifically, for a sensor array containing a large number of individual sensors, and assuming that it is desirable to capture a large number of frames per second, then in the interval between consecutive frame captures the system needs to be able to store the values captured at each sensor, and transmit all of them out to a processor for image processing tasks and compression. If the maximum bandwidth in reading out from the sensor array is limited, this can mean that as the frame rate increases the maximum frame resolution would have to decrease. In these systems the data captured by each sensor would be first quantized and then transmitted to the processor.

Distributed coding techniques can play a role in this context as they enable reduction the number of bits transmitted by each individual sensor. As an example, the sensors can be divided into subsets, e.g., sensors of a given class belonging to a specific lattice within the sensor array. Then the sensor data can be organized into bit planes. For a given bit plane, the complete bit plane information is extracted from the array unchanged and sent to the processor. For other classes of pixels, the corresponding bit plane information is assumed to be correlated to the previously encoded bit plane and thus, rather than being sent in its entirety, only syndrome information corresponding to a pre-specified code is transmitted. The amount of information to be sent as syndrome depends on the level of correlation between the two classes of sensors, which can be expressed, for each bit plane, as a probability of error.